UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

TRABAJO FIN DE GRADO

Reconstrucción de fondo de escena basada en la mediana

Emilio Gómez García.
Tutor: Diego Ortego Hernández.
Ponente: Jose María Martínez Sánchez.

Julio 2017

Reconstrucción de fondo de escena basada en la mediana

Emilio Gómez García
Tutor: Diego Ortego Hernández
Ponente: Jose María Martínez Sánchez

Video Processing and Understanding Lab
Departamento de Tecnología Electrónica y de las Comunicaciones
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio 2017

Trabajo parcialmente financiado por el Ministerio de Economía y Competitividad del Gobierno de España bajo el proyecto TEC2014-53176-R (HAVideo) (2015-2017)

Resumen

En este TFG se aborda la tarea de reconstrucción o inicialización de fondo centrándose en el análisis del funcionamiento del algoritmo LabGen-P y en algunas variantes propuestas del mismo. Este algoritmo se basa en reconstruir el fondo usando la mediana temporal de los *pixels* con menos movimiento. La selección de dicho algoritmo se debe a sus buenos resultados en la reciente competición sobre inicialización de fondo[1].

Inicialmente, se estudiará el estado del arte para comprender los distintos retos que plantean la tarea de inicialización de fondo y qué estrategias se han utilizado en la literatura para abordarlos. Después se procederá a la implementación del algoritmo LabGen-P, cuyos resultados son de una alta calidad, con el objetivo de comprender su funcionamiento, bondades y deficiencias. A continuación, se introducirán modificaciones encaminadas a solventar dichas deficiencias, entre las que están la estimación de movimiento empleada y la inclusión de una nueva información espacial basada en el tamaño de las regiones.

Finalmente, se evaluará el rendimiento del algoritmo base y cada una de las modificaciones introducidas en él, con la finalidad de analizar sus funcionamientos y determinar cuál es el mejor de ellos.

Palabras clave

Inicialización de fondo, mediana temporal, fondo, frente, estimación de movimiento, regiones, flujo óptico.

[1] http://scenebackgroundmodeling.net/

Abstract

In this Bachelor´s thesis we have tackled the task of reconstructing or initializing a background image using the LabGen-P algorithm and some proposed variants. This algorithm reconstructs the background using the temporal median of the pixels with less motion in time. This algorithm has beeen selected due to the good results obtained in a recent background initialization competition[2].

Initially, we will study the state-of-the-art to fully understand the different challenges that background initialization poses and the techniques used in the literature to addressed them. Subsequently, we will implement the LabGen-P algorithm, which has high quality results, in order to understand its strengths and weaknesses. Then, we will introduce variations in the algorithm to deal with its issues. Among these variations we can find the use of an improved motion estimation and the inclusion of spatial information based on segmented image regions sizes.

Finally, we will evaluate the base algorithm and the proposed variations to determine the best configuration.

Keywords

Background initialization, temporal median, background, foreground, motion estimation, regions, optical flow.

[2] http://scenebackgroundmodeling.net/

Agradecimientos

En primer lugar, quiero dar las gracias a Diego, mi tutor, por el apoyo que me ha ofrecido durante la duración del TFG, ayudándome siempre que lo he necesitado y explicándome todo las veces que fueran necesarias, siempre con buena cara. Por ello, gracias por su paciencia y por darme la oportunidad de realizar este proyecto... sin olvidarme de los miembros del VPU que tanto a mí, como a mis compañeros nos han acogido como a uno más.

También quiero agradecer a todos los amigos que me han ido surgiendo durante la carrera: Serrita, Pol, Míguel, Julianin, Sergy, Jose, User... Además de las últimas incorporaciones al grupo, que raramente son mujeres: Ana, Clau, Julia; Paula... Gracias a todos, estos 5 años han sido mejores gracias a vosotros, dejando grandes momentos que no olvidaré nunca, pero también mucho sufrimiento.

Gracias a todos mis amigos, especialmente a mis amigos de toda la vida, los KAMPEROS, con ellos empecé siendo un niño y ahora estoy hecho todo un hombretón. Hemos pasado momentos increíbles juntos, sin duda, de los mejores de mi vida, por eso gracias por ser como sois y por estar ahí siempre que lo he necesitado. Principalmente gracias a Charly, desde que teníamos 4 años ha sido como un hermano para mí, y a Paulita, una de mis pocas confidentes.

Finalmente, quiero agradecerle a mi familia su apoyo, especialmente a mi hermana y a mis papis, por su incondicional apoyo y confianza, además de darme esta oportunidad de realizar mis estudios, muchísimas gracias.

Pero especialmente quería agradecerle todo lo que ha hecho por mí a mi abuelita, la cual desafortunadamente nos ha dejado hace pocos días. Por ello quiero dedicarle, no solo este trabajo, sino todo el esfuerzo que he dedicado durante estos 5 años a superar la carrera. Te lo dedico a ti abuela, te quiero.

Gracias a todos.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

1.1. Motivación

En la actualidad, se ha incrementado la importancia de algoritmos de tratamiento de imagen debido a su gran avance tecnológico y su bajo coste, como puede ser en aplicaciones de interacción hombre-máquina o en muchas aplicaciones de visión artificial. El análisis automático de secuencias de video-vigilancia también es un importante área de investigación, por ello el uso de este tipo de algoritmos se está haciendo indispensable en la mayoría de entornos públicos que cuentan con gran afluencia de gente, ya sea en aeropuertos, carreteras, estaciones, centros comerciales u otros lugares de tránsito. Este continuo tránsito de personas u objetos en las escenas de análisis, hacen que la configuración de este tipo de algoritmos sea muy dependiente de la escena y sus condiciones.

La sustracción de fondo o *background subtraction*[1] es una de las etapas más importantes de pre-procesamiento en muchas aplicaciones de visión artificial o video-vigilancia, en las que se necesita detectar movimiento, identificar y/o seguir objetos en vídeos, etc. Los algoritmos de sustracción de fondo suelen emplearse cuando se dispone de una cámara fija, realiza una segmentación entre los objetos de frente y los de fondo, mediante la comparación del *frame* bajo análisis y un modelo de fondo de la escena. El modelado de dicho fondo supone un reto para los algoritmos de sustracción de fondo debido a sus variaciones espacio-temporales (cambios de iluminación o desaparición de objetos) a las que el algoritmo debe adaptarse. Por tanto, inicializar o re-inicializar el modelo de fondo es una tarea necesaria para mejorar su rendimiento, siendo la inicialización de fondo o *background initialization* (BI) [2] una respuesta a este problema.

La tarea de BI consiste en la estimación del fondo de la escena a partir de algunos

frames del vídeo. A pesar de los numerosos avances en la literatura para mejorar los resultados de BI [2], no hay algoritmos que funcionen bien en todos los entornos, tal y como se concluyó en la reciente competición de Diciembre de 2016[1]. Algunos de los entornos que propician dificultades para la BI los podemos observar en la Figura 1.1.

Por lo tanto la motivación de este TFG, es comprender el algoritmo ganador de dicha competición [3], un algoritmo sencillo que se basa en la mediana temporal, para analizar sus resultados e introducir modificaciones que sean capaces de aumentar la calidad de los fondos generados.

(a) (b) (c) (d)

Figura 1.1: Ejemplo de inicialización de fondo. Las imágenes (a), (b) y (c) se corresponden con *frames* de la secuencia de vídeo donde se pueden observar objetos estáticos y en movimiento que producen una fuerte oclusión del fondo. La imagen (d) se corresponde con el fondo de dicha secuencia que un algoritmo de inicialización debería generar.

1.2. Objetivos

El objetivo de este TFG es el análisis y mejora del algoritmo de inicialización de fondo [3]. Dicho objetivo se divide en los siguientes sub-objetivos:

1. Estudio del estado del arte. El objetivo de este apartado es familiarizarse con las técnicas empleadas en la literatura para realizar la inicialización de fondo, pudiendo así adquirir una visión amplia del campo.

2. Estudio del algoritmo base [3]. El objetivo de este apartado es comprender dicho algoritmo base mediante su implementación en MATLAB.

3. Mejoras del sistema base. El objetivo de este apartado es añadir modificaciones al algoritmo original que permitan mejorar la calidad de la inicialización de fondo

[1] http://scenebackgroundmodeling.net/

4. Análisis de resultados. En este apartado el objetivo es comparar las distintas modificaciones realizadas con el algoritmo base y otros algoritmos de la literatura en *datasets* públicos y privados para analizar los resultados obtenidos

1.3. Organización de la memoria

La memoria consta de los siguientes apartados:

- Capítulo 1. Introducción 1. Motivación del trabajo y objetivos.
- Capítulo 2. Estado del arte 2. Estado del arte sobre la estimación de fondo, sistemas previos y retos.
- Capítulo 3. Algoritmo seleccionado 3. Estudio del algoritmo inicial, sobre el que se desarrollarán algunas modificaciones, con el objetivo de superar algunos de los retos de la inicialización de fondo.
- Capítulo 4. Evaluación 4. Información sobre los *datasets* y las métricas empleadas para su evaluación. Evaluación comparativa de los distintos resultados obtenidos.
- Capítulo 5. Conclusiones y trabajo futuro 5. Razonamientos personales y mejoras que se podrían introducir en estos sistemas.
- Bibliografía.

Capítulo 2

Estado del arte

2.1. Introducción

La sustracción de fondo [1] es una técnica ampliamente utilizada para segmentar objetos de frente o *foreground* en entornos con cámaras estáticas o cámaras móviles en las es asumible modelar el fondo de escena. En el campo de la video-vigilancia, es indispensable ya que permite segmentar rápidamente objetos de interés para llevar a cabo modelados de las situaciones que tienen lugar en la escena monitorizada. Segmentar los objetos de interés consiste en generar máscaras binarias de las regiones de frente mediante comparaciones entre cada *frame* analizado y un modelo de fondo de la escena. Los algoritmos de BS pueden dividirse en cuatro etapas: Inicialización de fondo [2], su objetivo es generar una imagen inicial de fondo a partir de unos pocos *frames* del vídeo; Modelado de fondo, intenta capturar la evolución de la escena a lo largo del tiempo, representando estadísticamente el fondo; Actualización de fondo, debido a la variabilidad temporal del fondo de la escena, es necesario que este fondo se actualice con el paso del tiempo para no quedar obsoleto**;** Detección de frente, consiste en la detección de los objetos que no son fondo, mediante la comparación del modelo de fondo con los *frames* entrantes.

Este trabajo se centra en la primera de las etapas, la inicialización de fondo, que en el contexto de sustracción de fondo encuentra su utilidad en servir de información inicial fiable para el modelado ya sea al principio de un vídeo o en instantes posteriores en los que quiera re-inicializarse el fondo al haber quedado obsoleto. Por otro lado, BI es una tarea importante no solo en el contexto de segmentación de objetos sino en diferentes aplicaciones:

- Segmentación de vídeo. Se utiliza el fondo generado para la extracción de los objetos que forman el frente del vídeo, de esta forma conseguimos diferenciar

foreground y *background.*

- Restauración de vídeo. El fondo ayuda a reparar partes deterioradas de algún *frame* de la secuencia.
- Compresión de vídeo. El fondo representa información redundante de la escena, por lo que conociéndolo se puede evitar enviar información innecesaria.
- Privacidad de personas. El fondo es una información que puede explotarse para preservar la privacidad de las personas.
- Fotografía computacional. El fondo proporciona una imagen libre de objetos, que es lo que busca el usuario a partir de un conjunto de *frames* que contienen objetos de frente.

A pesar de la importancia de la inicialización de fondo, se trata de un campo que no posee tanta investigación como el resto de etapas de BS. Esta etapa de inicialización, es el punto de partida de los algoritmos de BS, su objetivo es la extracción de una imagen de fondo de escena libre de objetos. Generalmente, los algoritmos de BS emplean sistemas de BI sencillos, asumiendo que el fondo de la escena se obtiene fácilmente a partir de unos pocos *frames* del vídeo. Sin embargo, esta suposición puede no cumplirse en muchos escenarios de video-vigilancia, como en centros comerciales, donde pueden aparecer gran cantidad de objetos estáticos o grandes y continuas oclusiones del fondo de escena, dificultando su estimación. En definitiva, la tarea de BI es una tarea compleja, con gran cantidad de retos definidos (ver Subsección 2.3), que dificultan su obtención.

2.2. Sistemas previos

La estrategia de la mayoría de algoritmos BI para estimar una imagen de fondo es asumir que el fondo es homogéneo y que no posee movimiento, es estático. El concepto de inicialización de fondo, recibe diversos nombres en la literatura [4, 5]: *Bootstraping* [6, 7], BG *estimation* [8, 1], BG *generation* [9, 10] o BG *reconstruction* [11]. Los métodos de inicialización de fondo se pueden organizar según el método en el que se basen [2]: *Temporal Statistics, Sub-intervals of stable intensity, Iterative Mode Completion* u *Optimal Labeling*. Sin embargo, se va a utilizar la clasificación propuesta en [12] donde los algoritmos se clasifican en función de si la estrategia aplicada se basa en características temporales (basados en la idea de que el fondo no posee movimiento) o espaciales (basados en la idea de que el fondo es homogéneo).

La información temporal y espacial se puede usar en modo *batch* o en un análisis *online*, operando a nivel de píxel o de bloque.

Las aproximaciones que emplean estrategias temporales están presentes en muchos algoritmos de BS [2] donde el fondo se construye a partir de una actualización del primer *frame* de la secuencia [7]. Sin embargo, estas aproximaciones se basan en la mayoría de los casos, en la suposición de que el fondo es visible la mayoría del tiempo, motivo por el cual los resultados no siempre son precisos. La mediana es una alternativa empleada en algunos trabajos [13, 14], pero es una estrategia que falla cuando los objetos de frente están estáticos más del 50 % del tiempo, puesto que son considerados como parte de fondo. No obstante, existen variaciones recientes basadas en la mediana [3] que filtran *pixels* no interesantes para el cálculo de la mediana haciendo uso de información de movimiento, consiguiendo resultados muy buenos.. Además, algunos algoritmos emplean estimación de movimiento para evitar que objetos del frente pasen a formar parte del fondo de la secuencia, mediante estimaciones de movimiento con flujo óptico u *optical flow* (OF) [15] y diferencias entre imágenes o *frame difference* [6, 9]. La estabilidad temporal también se utiliza [15] para obtener varios candidatos a representar el fondo en cada región espacial. Sin embargo, dicha estabilidad no es capaz de agrupar correctamente los candidatos que representan el fondo, ya que supone que los intervalos no continuos de tiempo modelan representaciones distintas del fondo. Por lo tanto, se prefiere emplear técnicas de *clustering* que tienen en cuenta similitudes entre intervalos no continuos [4, 8, 16].

Aunque algunas aproximaciones solo emplean un análisis temporal [17], es necesario incluir también un análisis espacial para lidiar con la suposición errónea de que el fondo es temporalmente dominante en la secuencia. Es habitual emplear criterios basadas en el color para decidir si un píxel pertenece al frente o al fondo que estaba previamente ocluido [9]. En [8], extendido en [16], la continuidad espacial se modela a nivel de bloque usando la Transformada Discreta del Coseno o *Discrete Cosine Transform* (DCT) mediante un MRF (*Markov Random Field*), junto con un módulo iterativo que corrige posibles errores. Con el objetivo de reducir la complejidad computacional de la DCT, en [18] se emplea la Transformada de Hadamard junto con una corrección del fondo basada en descartar bloques en los que el gradiente en sus bordes es alto. En [4], se utiliza un esquema de bloques solapados donde la continuidad espacial reside en las diferencias de color (entre áreas no solapadas) y en la distancia *chi-square* (entre áreas solapadas) entre el bloque candidato a ser fondo y el fondo ya establecido. Además, las relaciones de color y gradiente entre bloques vecinos se utilizan en [19] para construir el fondo de la secuencia. Recientemente, [12] realiza una generación del fondo basada en continuidades espaciales multi-camino, es

decir, llegando a los bloques a reconstruir desde distintos caminos espaciales. También existen aproximaciones que agrupan la información de continuidad espacial junto con información temporal para llevar a cabo una minimización de energía usando *Loopy Belief Propagation* [20, 21], *Graph Cuts* [1], *Conditional Mixed-State* MRF [11] o *Dynamic* MRF [5]. Finalmente, el uso del *optical flow* en posiciones vecinas realizado por [15] puede considerarse también como información espacial.

Se han propuesto varias estrategias para afrontar la tarea de la inicialización de fondo, sin embargo, ninguna de ellas es capaz de operar correctamente en todas las situaciones debido a la cantidad de retos (ver Sección 2.3) a los que se enfrentan. No obstante, puede apreciarse un cierto consenso en torno a la idea de que la homogeneidad y estaticidad del fondo son los criterios que proporcionan mejores rendimientos en la tarea de BI.

2.3. Retos de la inicialización de fondo

La inicialización de fondo[2] plantea diferentes retos a superar debido a la complejidad y heterogeneidad de situaciones que pueden acontecer en las secuencias de vídeo. A continuación, se presentan distintos factores que dificultan la tarea de inicialización de fondo [12]:

1. Sombras y reflejos. Una sombra es una región en la que al ser obstaculizada la luz, se origina una disminución de luminancia; mientras que un reflejo es un aumento de la luminosidad en una región debido a la incidencia de luz y a las propiedades del propio material. El movimiento de objetos en la escena provoca la aparición de estos fenómenos (ver Figura 2.6 (a)), que introducen cambios de luminancia en la escena que deben ser evitados a la hora de estimar el fondo.

2. Fondo dinámico. Existen elementos que pueden estar presentes en los vídeos asociados al fondo de escena pero que tiene asociado un movimiento (p.ej. árboles, agua o escaleras mecánicas) (ver Figura 2.6 (b)) y por lo tanto dificultan una separación entre frente y fondo basada en el movimiento.

3. Camuflaje. Cuando los objetos de frente tienen un color similar al fondo tiene lugar una situación de potencial camuflaje (ver Figura 2.6 (c)), donde los algoritmos de inicialización que utilicen la premisa de fondo homogéneo pueden fallar al existir una homogeneidad entre el frente y el fondo.

4. Ruido. El ruido introducido en una secuencia, generalmente producido en la captación de la secuencia de vídeo (ver Figura 2.6 (d)), puede provocar grandes

Figura 2.1: Ejemplo de secuencia con oclusión de fondo. Las regiones de fondo no son visibles durante gran parte de la duración de la misma.

errores en la estimación del fondo del vídeo, tanto en la segmentación de los objetos del vídeo como en la estimación de movimiento.

5. Cambios de iluminación. A lo largo del tiempo, el fondo sufre variaciones de iluminación graduales y bruscas (ver Figura 2.3) que llevan a que la imagen de fondo correcta no sea fija, pues existen varios fondos válidos.

6. Objetos estáticos. Este es uno de los problemas más destacados en la inicialización de fondo, su complicación reside en la aparición de objetos de frente que o bien tras poseer un movimiento en la escena se quedan parados o bien tras estar parados empiezan a moverse (ver Figura 2.2) y por lo tanto al carecer de movimiento no permiten realizar una separación frente-fondo basada en asumir que el fondo es estático.

7. Oclusión continuada del fondo. Es habitual que en las escenas a analizar haya oclusiones continuadas del fondo de escena (ver Figura 2.1) debido a objetos en movimiento que dificultan su visualización y por tanto su generación.

8. *Jitter*. Este efecto sucede cuando la cámara de vídeo sufre pequeñas vibraciones debidas, generalmente, al viento. Este efecto provoca que se detecte movimiento en toda la escena, dificultando la estimación del fondo mediante el movimiento de la misma. En la Figura 2.4, se puede apreciar este efecto, como el emborronamiento de los objetos de la escena debido al movimiento de la cámara.

9. Secuencias de corta duración. Cuando se dispone de secuencias con poca cantidad de *frames*, la estimación de movimiento se ve comprometida debido a los cambios bruscos que se producen entre *frames* próximos que llevan a disponer de poca información para inicializar el fondo. En la Figura 2.5 se aprecia este violento cambio de escena entre imágenes consecutivas.

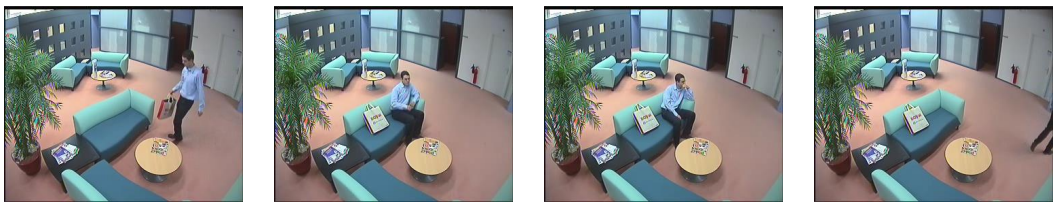


Figura 2.2: Ejemplo de secuencia con objetos estáticos. La bolsa que deposita el hombre en el sofá, permanece estática más del 50 % del tiempo.

Figura 2.3: Ejemplo de secuencia con cambios de iluminación. Se produce un cambio de iluminación drástico de una imagen a otra.

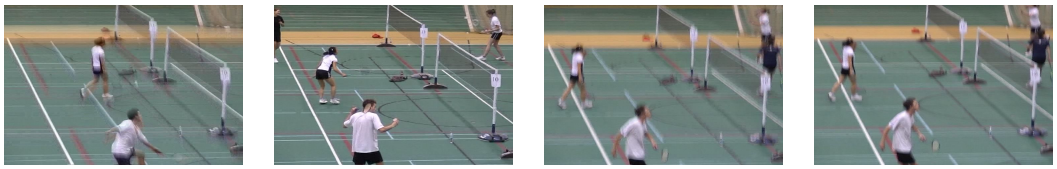


Figura 2.4: Ejemplo de secuencia con *jitter*. Este efecto se puede apreciar en el emborronamiento de las imágenes.

Figura 2.5: Ejemplo de secuencia de corta duración. Las imágenes se corresponden con *frames* consecutivos.

(a) (b) (c) (d)

Figura 2.6: Ejemplo de retos para la inicialización de fondo. La imagen (a) hace referencia al reto de sombras y reflejos. La imagen (b) acomete el reto del fondo dinámico. La imagen (c) aborda el reto del camuflaje. La imagen (d) muestra el reto del ruido.

Capítulo 3

Algoritmo seleccionado

3.1. Introducción

Para trabajar en la inicialización de fondo se ha decidido seleccionar un algoritmo reciente de la literatura que obtenga buenos resultados para implementarlo en MATLAB, analizarlo e incluir modificaciones que aborden sus problemas. El algoritmo seleccionado es LabGen-P [3], ganador de una competición sobre inicialización de fondo en el año 2016[1] gracias a su buen funcionamiento en la mayoría de entornos. No obstante, a pesar de su buen funcionamiento, esta estrategia obvia la problemática que introducen los objetos completamente estáticos que entran o abandonan la escena durante los *frames* analizados. En la Sección 3.2 se introduce el algoritmo base [3], mientras que en la Sección 3.3 se presentan las modificaciones realizadas con el objetivo de mejorar el algoritmo base.

3.2. Algoritmo base

3.2.1. Esquema general

El algoritmo base [3] genera una imagen de fondo de una secuencia de vídeo mediante la combinación de la estrategia de la mediana a nivel de píxel y la estimación de movimiento entre *frames* consecutivos. Se basa en el algoritmo inicial LabGen [22], que lleva a cabo el análisis a nivel de bloque en vez de a nivel de píxel, mediante el cual se obtenían buenos resultados pero con un efecto bloque importante en ciertos entornos. El algoritmo base está estructurado en las siguientes etapas:

[1] http://scenebackgroundmodeling.net/

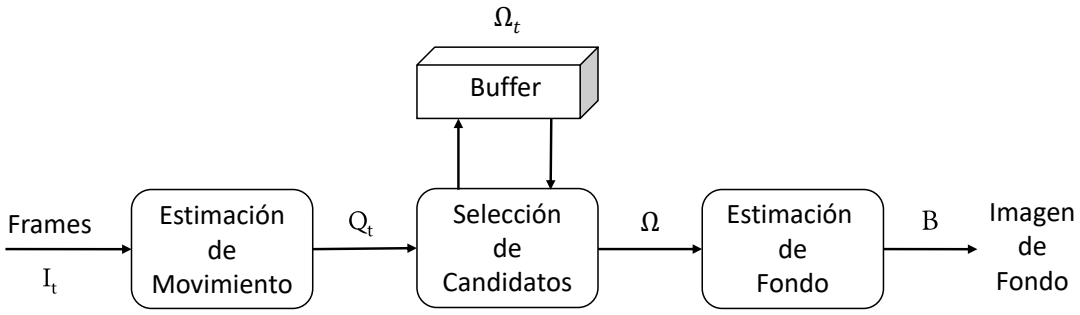


Figura 3.1: Esquema del algoritmo base. Donde I_t se corresponde con una imagen del vídeo en escala de grises, Q_t y $\Omega_t$con la puntuación de movimiento y el estado del *buffer* en ese instante respectivamente y siendo Ω el estado final del buffer y B la imagen de fondo obtenida.

1. Estimación de movimiento. A partir del *frame* actual y el anterior determina en cada instante una puntuación de movimiento Q_t empleada para diferenciar zonas potencialmente del fondo y del frente de la escena.

2. Selección de candidatos. Se almacenan en un *buffer* Ω los *pixels* con menor movimiento a lo largo de la secuencia, de forma que en cada instante se sustituyen aquellos *pixels* almacenados que tengan un movimiento mayor que el de los *pixels* actuales por dichos *pixels* actuales. El tamaño del *buffer* en cada posición viene definido por S, es decir, el número de *pixels* que se guarda en cada posición.

3. Estimación de fondo. Finalmente el fondo se estima mediante la aplicación de la mediana temporal sobre cada uno de los conjuntos de S *pixels* almacenados en el *buffer* Ω, obteniendo así la imagen de fondo estimada B.

3.2.2. Descripción

Con el objetivo de diferenciar entre frente y fondo, el algoritmo realiza un análisis comparativo entre cada *frame* y su homólogo anterior para estimar el movimiento. Los resultados obtenidos en esta estimación de movimiento quedan definidos por una puntuación o *score* de movimiento Q_t para cada *frame* I_t, donde I en este apartado t es el subíndice que denota el instante temporal. Cada píxel p de la imagen queda definido por I_t^p, siendo p un vector con dos dimensiones, de manera que Q_t es el conjunto de puntuaciones para cada píxel de la imagen p y su cálculo puede expresarse como una función f tal que:

$$Q_t = f\left(I_t, I_{t-1}\right), \tag{3.1}$$

donde el movimiento en el instante t depende del *frame* actual I_t y el anterior I_{t-1}. Esta función de movimiento LabGen-P la define como un *frame difference* a partir de las imágenes en escala de grises expandido espacialmente debido a que esta estimación es una información con variación mayoritaria en los bordes, ya que al obtenerse mediante la resta de *frames* consecutivos solo detecta movimiento en estas posiciones. Por tanto, f queda definido como:

$$f\left(I_t, I_{t-1}\right) = \sum_{p \in \mathcal{M}} \left|I_t^p - I_{t-1}^p\right|, \tag{3.2}$$

siendo $\mathcal{M}$ un vecindario rectangular de lado $W = 1 + 2 \times \left[\frac{min(\#filas, \#columnas)}{2*N}\right]$, I una imagen de luminancia y N una constante. Esta estimación de movimiento a partir de puntuaciones, evita la necesidad de implantar un umbral para discernir entre fondo y frente.

Una vez calculada la información de movimiento Q_t, mediante el paso de selección de candidatos, se introduce dicha información de manera iterativa para cada píxel I^p en un *buffer* Ω^p, generando subconjuntos de un máximo de S *pixels*, utilizando un criterio de puntuación mínima de movimiento para introducir las puntuaciones de cada píxel Q_t^p en este espacio. Con el objetivo de inicializar el *buffer*, las puntuaciones de los primeros S *pixels* Q^p son introducidas directamente en Ω^p. Una vez que el *buffer* se encuentre completo, el píxel únicamente será añadido de la siguiente manera:

$$\Omega_t^p\left(d\right) = \begin{cases} I_t^p & \exists d : argmax_d\left(Q_t^p - \Omega_{t-1}^p\left(d\right)\right) < 0 \\ \Omega_{t-1}^p\left(d\right) & resto \end{cases}, \tag{3.3}$$

donde d es la dimensión de Ω^p, tomando valores en el rango $[1, S]$. Mediante este criterio se comprueba si la cantidad de movimiento Q_t^p asociada a un píxel de la imagen I_t^p, es menor o igual a la cantidad de movimiento de un píxel perteneciente al buffer Ω^p. En caso de ser menor este nuevo píxel I_t^p se reemplaza por aquel píxel del conjunto $\{\Omega_t^p\left(d\right)\}_{d=1}^S$ que posea una mayor puntuación de movimiento. En el caso de que varios *pixels* de Ω^p tengan la misma puntuación máxima, se reemplazará aquel que sea más antiguo.

Una vez procesados todos los *frames* de la secuencia, el último paso consiste en la obtención del fondo B de la escena mediante aplicación de la mediana temporal a nivel de píxel sobre el *buffer* Ω^p. También existe la posibilidad de obtener una imagen de fondo en un instante t dado, mediante la aplicación de dicha mediana a cada Ω_t^p.

Por lo tanto, para calcular el valor de cada píxel B_t^p en dicha imagen de fondo se utiliza una función g tal que:

$$B_t^p = g\left(\Omega_t^p\right), \tag{3.4}$$

donde la imagen de fondo en el instante t depende del contenido del buffer en ese instante Ω_t^p. Esta función de movimiento LabGen-P la define como la mediana temporal. Por tanto, g queda definido como:

$$g\left(\Omega_t^p\right) = \text{med}\left(\left\{\Omega_t^p\left(d\right)\right\}_{d=1}^{S}\right), \tag{3.5}$$

siendo $\text{med}\left(\cdot\right)$ la mediana de un conjunto de valores. En cuanto a los parámetros utilizados para el algoritmo son $S = 19$ y $N = 3$, tal y como se define en [3].

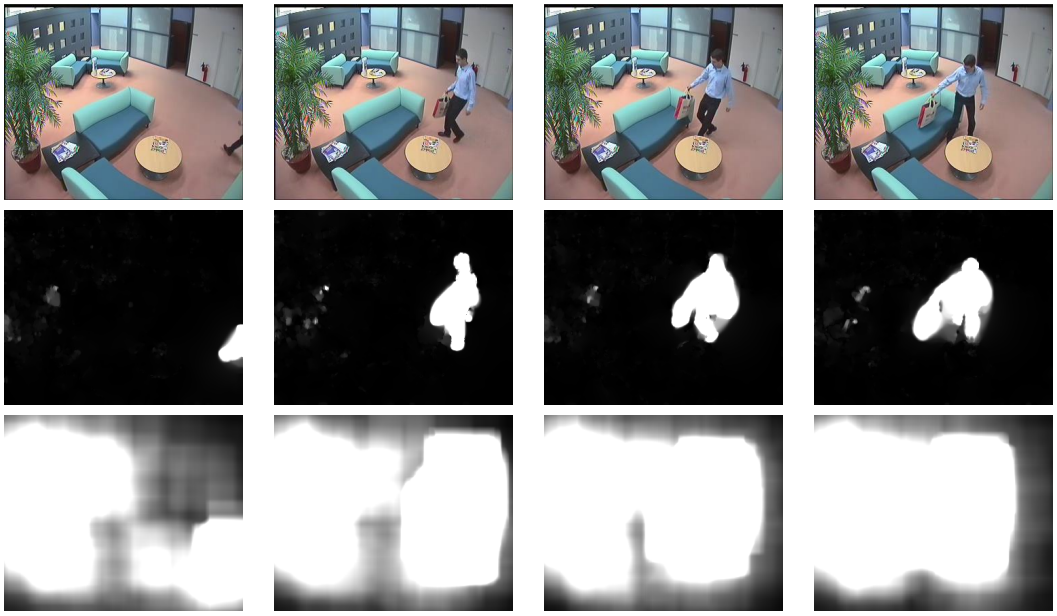
3.3. Modificaciones del algoritmo base

A partir de la implementación del algoritmo base se analizaron sus deficiencias (ver resultados cualitativos y cuantitativos en la Sección 4.2.1) para llevar a cabo modificaciones destinadas a lidiar con dichos problemas. En este apartado se incluyen modificaciones realizadas sobre el algoritmo base, con el objetivo de mejorar algunos de sus puntos débiles: la estimación de movimiento y la ausencia de estrategia para enfrentarse a los objetos estáticos

3.3.1. Estimación de movimiento

La estimación de movimiento mediante *frame difference* no es una estimación precisa del movimiento tal y como se puede ver en la Figura 3.2, ya que al basarse en la resta de *frames* consecutivos solo obtiene movimiento en los bordes de los objetos que se encuentran en movimiento y no en todo el objeto, por ello posteriormente se realiza una expansión (en todas direcciones y sin considerar las propiedades espaciales de la imagen) de dicho movimiento con el objetivo de abarcar los objetos completamente y no solo sus bordes. De esta forma, se obtiene un estimación de movimiento burda que no se adapta al tamaño del objeto en movimiento, sino que lo amplia enormemente, otorgando puntuación de movimiento a la mayoría de *pixels* de la imagen, aunque la mayoría de ellos carezcan de movimiento.

Una estimación más precisa puede obtenerse a partir de un algoritmo de flujo óptico u *optical flow* [23], que proporciona vectores de movimientos a nivel de píxel teniendo en cuenta las propiedades espaciales de la imagen (gradientes de color, suavidad espacial o escalas). Dichos vectores de movimiento, representados por $\mathcal{O}$,



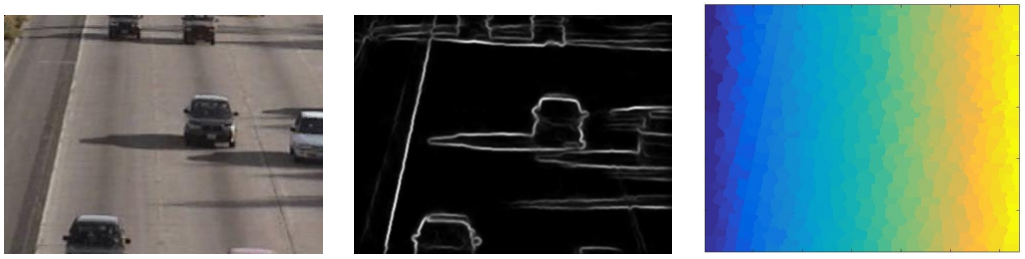
(a) (b) (c) (d)

Figura 3.2: Ejemplo de la estimación de movimiento con flujo óptico y con *frame difference*. De arriba a abajo, la primera fila se corresponde con imágenes de la secuencia, la segunda con sus estimaciones obtenidas mediante flujo óptico y la tercera mediante *frame difference*. En las estimaciones, las zonas claras implican una mayor detección de movimiento que las zonas oscuras.

indican el desplazamiento $\mathcal{O}(c)$ en cada dimensión c de la imagen (filas y columnas) entre dos *frames* analizados. . Para ello, en cada instante t se utiliza como información de movimiento Q_t^p para cada píxel p, el módulo del flujo óptico:

$$Q_t^p = \|\mathcal{O}_t^p\|\,, \tag{3.6}$$

donde $\|\cdot\|$ es la norma euclídea. En la Figura 3.2 puede apreciarse la diferencia de calidad entre el movimiento del algoritmo original y el flujo óptico empleado, siendo la estimación del *frame difference* muy burda al obtener una elevada puntuación de movimiento en la mayoría de regiones de la imagen cuando en realidad el movimiento solo se encuentra en una pequeña zona de ella. Sin embargo, el flujo óptico consigue una puntuación mucho más ajustada al movimiento real que se produce en la escena, lo que puede ayudar a obtener una estimación más precisa de los objetos de frente en situaciones complejas, como aquellas con gran oclusión del fondo por objetos de frente. Se ha decidido emplear el algoritmo de flujo óptico propuesto en [23] ya se utiliza ampliamente en la literatura [24, 25] y tiene un buen balance entre precisión y velocidad.



(a) (b) (c)

Figura 3.3: Ejemplo de bordes y *superpixels* propuestos por Piotr Dollar. Donde (a) es un *frame* de un vídeo, (b) son sus bordes obtenidos y (c) sus *superpixels*.

3.3.2. Objetos estáticos

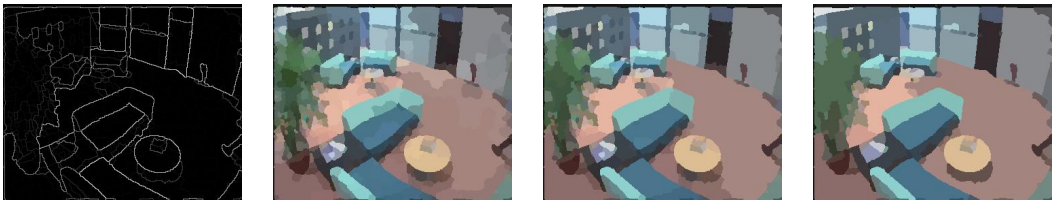
Como se ha visto, la estimación de fondo de LabGen-P se basa en realizar la mediana del conjunto de píxeles (*buffer* Ω) con menor movimiento asociado. Dicha estrategia no considera los objetos estáticos, es decir, objetos que tras un movimiento previo se quedan parados o que tras estar parados empiezan a moverse y por tanto, al no tener movimiento, pueden formar parte de Ω.

Tradicionalmente, en el estado del arte se emplean estrategias de continuidad espacial para estimar el fondo [16, 12]. Estas estrategias son robustas a los objetos estáticos pero tienen problemas para lidiar simultáneamente con situaciones de elevado movimiento. Para introducir información espacial en la estrategia de inicialización de LabGen-P, se ha definido una nueva puntuación Q_t que depende de información espacial. En particular, se ha realizado una segmentación en regiones basada en *superpixels* [26] para utilizar el tamaño relativo de las regiones en la escena cómo Q_t. La asunción detrás de utilizar este criterio se basa en que los fondos de la escena en muchas ocasiones se corresponden con regiones amplias mientras que los objetos tienen un tamaño menor.

Para obtener este tamaño de las regiones, en primer lugar se ha realizado una segmentación en *superpixels* de la imagen mediante una variación de [26] propuesta por Piotr Dollar[2] que se basa en color e información de bordes [27] (ver Figura 3.3).

En segundo lugar, se ha llevado a cabo un proceso de generación de regiones mediante la fusión de *superpixels* similares, con el objetivo de obtener regiones más grandes que se adapten a los objetos de la secuencia. Para lograr dicha fusión de *superpixels* se ha utilizado lo que se conoce como un *Ultrametric Contour Map* (UCM) [28]. El UCM es una imagen $\mathcal{U}_t$ donde están puntuados los bordes entre *superpixels* de manera que al umbralizar dichas puntuaciones y extraer componentes conexas,

[2] https://github.com/pdollar/edges/



(a) (b) (c) (d)

Figura 3.4: Ejemplo de UCM y distintas umbralizaciones del mismo. La imagen (a) se corresponde con el UCM de un *frame* de la secuencia y el resto son umbralizaciones del mismo con distintas intensidades: (b) con 0.01, (c) con 0.03 y (d) con 0.07.

se obtienen las regiones de la imagen. Para obtener esta imagen se lleva a cabo una segmentación jerárquica iterativa, en la que en cada iteración se fusionan regiones adyacentes en función de la intensidad de los bordes en la imagen de contornos calculada. De esta manera, en las primeras iteraciones se obtienen segmentaciones finas (*superpixels*) y según se va iterando se van fusionando más regiones, obteniendo así una segmentación más gruesa (objetos). Los valores de cada píxel p de $\mathcal{U}_t$ quedan definidas por el nivel de la iteración en el que desaparece cada borde entre *superpixels*, como se puede observar en la Figura 3.4.

Para obtener una segmentación en regiones $\mathcal{S}_t = \{S_{t,i}\}_{i=1}^{k}$, donde k indica el número de regiones $S_{t,i}$ de la imagen $\mathcal{I}_t$, se ha umbralizado $\mathcal{U}_t$ usando un valor de 0.03 para obtener las regiones, dado que empíricamente se ha comprobado que ese valor es capaz de mantener los objetos a la vez que fusiona partes homogéneas de los fondos. En la Figura 3.5 se pueden apreciar, en la primera fila, ejemplos del mapa UCM $\mathcal{U}_t$ obtenido en varios instantes de una secuencia, así como las respectivas regiones obtenidas al umbralizar con el valor seleccionado, en la segunda fila.

En tercer lugar, se ha establecido un criterio de diferenciación entre objetos de frente y fondo utilizando el tamaño de cada una de las regiones de la imagen. Esta puntuación es calculada de la siguiente forma:

$$Q_t^p = |S_i^p| \tag{3.7}$$

siendo S_i^p la región asociada al píxel p y$|\cdot|$el número de elementos.

En la Figura 3.6, se muestra la puntuación de tamaños de imágenes (b) y (d) sobre unos *frames* (a) y (c) de una secuencia con objetos estáticos. Al introducir el objeto estático en la escena (bolsa) la información de tamaños hace que al tener este objeto un tamaño inferior al fondo donde se sitúa, tenga una alta puntuación de ser un objeto de frente.

El objetivo es mejorar el resultado en situaciones con objetos estáticos, ya que

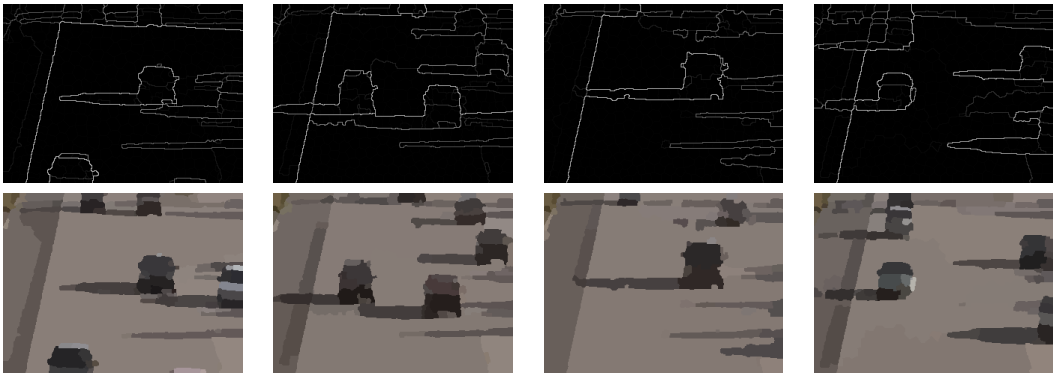
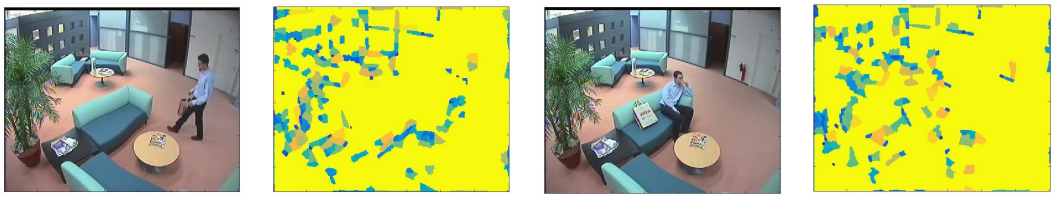


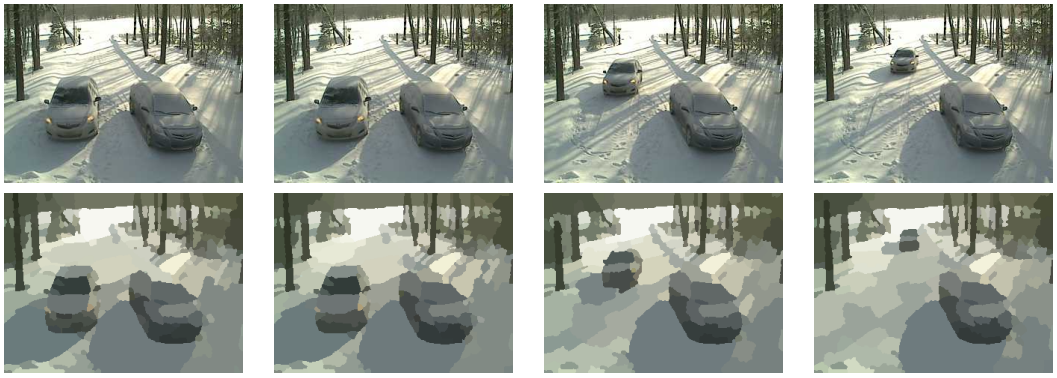
Figura 3.5: Ejemplo de regiones obtenidas en una secuencia y sus respectivos mapas UCM. La fila superior se corresponde con los mapas UCM obtenidos para distintos *frames* de una misma secuencia. La fila inferior se corresponde con las respectivas regiones obtenidas para cada *frame* a partir de dichos mapas de decisión empleando $\vartheta = 0{,}03$.



(a) (b) (c) (d)

Figura 3.6: Ejemplo de puntuación de tamaños de regiones. Las imágenes (a) y (c) se corresponden con *frames* de la secuencia y las imágenes (b) y (d) con sus respectivas puntuaciones de ser fondo obtenidas unicamente mediante la información de tamaño de regiones. Cuanto más oscuro sea el *score* (azul), más probabilidades tendrá de ser frente.

si estos son menores que las regiones de fondo no serán seleccionados a la hora de obtener la imagen de fondo final. No obstante, se produce el inconveniente de que cuando aparecen objetos de frente de un tamaño superior a los objetos de fondo, estos serán seleccionados como fondo de la escena. En la Figura 3.7 se puede apreciar un ejemplo de cómo se pueden solucionar algunos problemas de objetos estáticos con este método al tener que el tamaño de las regiones donde se encuentra el objeto estático es menor (ver Figura 3.7 (a) y (b)), mientras que cuando este se va, el tamaño de las regiones es mayor (ver Figura 3.7 (c) y (d)) y por tanto podrán seleccionarse dichos *pixels* asociados como *pixels* a incluir en el *buffer* Ω (ver Figura 3.8 (a)). En dicha figura también se aprecia la imagen de fondo (ver Figura 3.8 (b)) de la secuencia (ver Figura 2.1), en la cual no se obtiene el resultado deseado debido a que las regiones de frente son de mayor tamaño que las de fondo. Por ello, la información obtenida en



(a) (b) (c) (d)

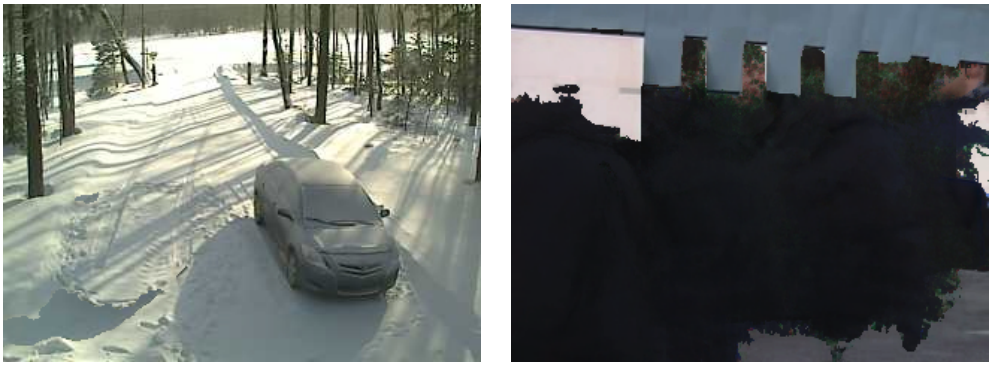
Figura 3.7: Ejemplo del análisis por regiones de una secuencia con objetos estáticos. La fila superior de la figura se corresponde con algunos *frames* del vídeo, mientras que la fila inferior se corresponde con la segmentación en regiones de estos *frames*. Los *frames* (a) y (b) se corresponden con la mayoría de *frames* del vídeo, mientras que (c) y (d) solo se corresponden con los instantes finales.

esta etapa necesita ser complementada para funcionar correctamente. En particular, en la Subsección 3.3.3 se va a presentar una forma de combinar la información de tamaño y movimiento.

3.3.3. Fusión de información de movimiento e información espacial

En este apartado se busca la unión de los dos conceptos vistos anteriormente, la información de movimiento y la de regiones. La información de movimiento Q_t^{Mov} (ver Figura 3.9 (fila superior (b))), si es calculada mediante *flujo óptico*, su expresión es la definida en la Ecuación 3.6 y si se emplea *frame difference*, esta información es obtenida a partir de la Ecuación 3.6. La información de tamaño de regiones Q_t^{Reg} (ver Figura 3.9 (fila superior (c))) se obtiene mediante la Ecuación 3.7. Con esta función, el objetivo es mantener los buenos resultados que se obtenían con la estimación de movimiento en el algoritmo base, introduciendo además la información de tamaño de regiones para intentar mejorar en aquellas situaciones en las que aparezcan objetos estáticos.

Por lo tanto reutilizaré los desarrollos empleados anteriormente, introduciendo una normalización de cada una de las dos informaciones al intervalo [0,1] para poder unificarlas, generando la puntuación de pertenencia al fondo Q_t mediante la suma de ambas. De esta forma, en el caso de que la estimación de movimiento se lleve a cabo mediante el *frame difference* (ver Subsección 3.2.2), el tamaño de ventana W seguirá siendo igual que el empleado en el algoritmo base , sin embargo si esta estimación se



(a) (b)

Figura 3.8: Ejemplos de fondos obtenidos empleando el análisis por tamaño de regiones. La imagen de la izquierda (a) se corresponde con el fondo obtenido empleando este análisis para la secuencia 3.7, mientras que la imagen de la derecha (b) es la obtenida para la secuencia 2.1.

lleva a cabo mediante el *flujo óptico* (ver Subsección 3.3.1), no será necesario su uso, ya que en este caso la estimación de movimiento se adapta correctamente a toda la región con movimiento.

La normalización de la puntuación de movimiento Q_t^{NMov} (ver Figura 3.9 (fila inferior (b))), difiere según el algoritmo empleado para su estimación:

Si se emplea *frame difference*, se usará la umbralización de Rossin para separar la información de movimiento $Q_t^{FD} = Q_t^{Mov}$ de la estática. Para no obligar a una separación en dos informaciones en aquellos casos en los que no exista movimiento, otorgando puntuación alta de movimiento a zonas estáticas, se propone el siguiente criterio:

$$\Phi' = max\left(3, \Phi\right), \tag{3.8}$$

donde max $(\cdot)$ es el máximo de ambos valores, Φes el umbral obtenido mediante el método de Rossin y se elige el valor 3 como límite mínimo del umbral, de esta forma cualquier puntuación por debajo de este valor nunca será catalogada con gran *score* de movimiento. Por lo tanto la normalización será:

$$Q_t^{NMov} = \begin{cases} 1 & Q_t^{FD} > \Phi' \\ \frac{Q_t^{FD}}{\Phi'} & resto \end{cases}. \tag{3.9}$$

Con el empleo del flujo óptico, el movimiento se ajusta perfectamente a los objetos de la escena (ver Figura 3.2 (fila 2)) y no es necesaria una umbralización para diferenciar entre frente y fondo. Se emplea una saturación brusca asignando *score*

máximo a aquellos valores superiores a 0.15, con el objetivo de anteponer la puntuación de movimiento $Q_t^{OF} = Q_t^{Mov}$ a la de tamaño de regiones. El valor de 0.15 ha sido elegido tras comprobar empíricamente que con esta saturación se asegura que un objeto en movimiento tenga la puntuación máxima. Para saturar se aplica la siguiente ecuación:

$$\mathcal{M}^t = \begin{cases} 1 & Q_t^{OF} > 0{,}15 \\ Q_t^{OF} & resto \end{cases}. \tag{3.10}$$

Una vez realizada dicha saturación, se realiza luna normalización de la siguiente manera:

$$Q_t^{NMov} = \frac{\mathcal{M}^t}{max(\mathcal{M}^t)}, \tag{3.11}$$

donde $\mathcal{M}$ es la información de movimiento saturada y $\max(\cdot)$ es la función máximo.

Para realizar la normalización de la información de tamaño de regiones Q_t^{NReg} (ver Figura 3.9 (fila inferior (c))), en el primer *frame* se calculan un valor máximo y mínimo de tamaño de regiones en la imagen. Estos valores son seleccionados como el percentil 3 Out_{Min} y el 97 Out_{Max} de un histograma que contenga todos los tamaños del primer *frame*. La saturación se emplea con el objetivo de eliminar tamaños aislados que empeoren la normalización, se realiza de la siguiente forma:

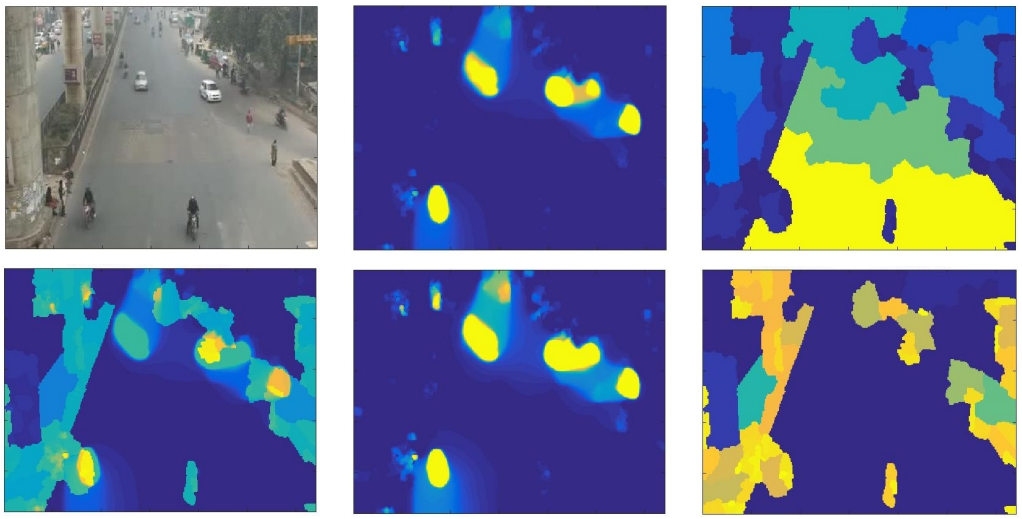
$$\mathcal{R}^t = \begin{cases} Out_{Max} & Q_t^{Reg} > Out_{Max} \\ Out_{Min} & Q_t^{Reg} < Out_{Min} \\ Q_t^{Reg} & resto \end{cases}. \tag{3.12}$$

Una vez realizada dicha saturación, se realiza luna normalización de la siguiente manera:

$$Q_t^{NReg} = \frac{\mathcal{R}^t}{max(\mathcal{R}^t)}, \tag{3.13}$$

Una vez obtenidas ambas informaciones normalizadas se procede a su unión. Para ello, primero es necesario invertir la puntuación de tamaños, ya que un valor pequeño de la misma implica frente, mientras que un valor pequeño de movimiento implica fondo. Por este motivo, $Q_t^{NReg} = 1 - Q_t^{NReg}$. La fusión de ambas informaciones se expresa con la siguiente ecuación:

$$P_{FG}^t = \lambda Q_t^{NMov} + (1 - \lambda)\, Q_t^{NReg}, \tag{3.14}$$



(a) (b) (c)

Figura 3.9: Ejemplo de algoritmo con información de movimiento y de tamaño de regiones. Si analizamos las imágenes de arriba abajo y de izquierda a derecha, la primera imagen se corresponde con un *frame* de la secuencia bajo análisis. La segunda figura es una representación del movimiento calculado mediante el flujo óptico. La tercera representación se corresponde con la información de tamaño de regiones (aún sin invertir). Las figuras que se encuentran debajo de estas dos últimas, son ellas mismas normalizadas (el tamaño de regiones ya invertido), para poder representar en la figura inferior izquierda la información de probabilidad de fondo como la suma de ambas, siendo aquellas zonas con colores más claros las que tienen mayor probabilidad de objetos de frente (amarillo) y aquellas zonas más oscuras las que se corresponden con el fondo (azul oscuro).

siendo Q_t^{NMov} y Q_t^{NReg} las informaciones de tamaño y movimiento normalizadas respectivamente y λun parámetro para otorgar más importancia a una u otra información, su valor es $\lambda = 0{,}5$.

Ahora P_{FG}^t (ver Figura 3.9 (fila inferior (a))) tendrá la misma función que Q_t en el algoritmo base (ver Sección 3.2). Por lo tanto, se utilizará un *buffer* similar con dicha información (ver Ecuación 3.3) con los mismos criterios de actualización. Para generar el fondo se realizará la mediana de dicho *buffer* (ver Ecuación 3.5).

En la figura 3.9 podemos apreciar un ejemplo de aplicación de este algoritmo, donde se obtienen mejores resultados que con el algoritmo original debido a la introducción de la información de tamaño de regiones. Se aprecia como en esta situación la utilidad que tiene la información de tamaño de regiones, ya la moto de la parte inferior de la secuencia no tiene puntuación de movimiento debido a que se corresponde con un objeto estático, pero gracias al uso de la información de tamaños, obtiene una puntuación alta de ser un objeto de frente.

Capítulo 4

Trabajo experimental

4.1. Marco de evaluación

4.1.1. *Datasets*

Para realizar la evaluación del algoritmo base realizado [3], así como la de todas aquellas modificaciones introducidas, se han empleado dos *datasets* diferentes, uno elaborado en el VPU (*Video Processing and Understanding Lab*) y otro aportado por el grupo de SBMnet (*Scene Background Modeling*)[1]. La elaboración de dichos *datasets* esta realizada con el objetivo de observar cómo se comporta el algoritmo de inicialización de fondo desarrollado frene a diferentes retos, como los explicados en la Sección 2.3. A continuación se presentan ambos *datasets,* que se encuentran estructurados en categorías.

VPU Dataset. Este *dataset* está estructurado en 4 subsecciones (ver Figura 4.1) , el principal objetivo es observar cómo se enfrenta el algoritmo a los retos de oclusión de fondo y objetos estáticos, para observar si con las modificaciones introducidas se consiguen mejoras. Estructuración del *dataset* realizado por VPU:

- *Baseline*. Categoría formada por secuencias sencillas, sin incluir grandes retos para la inicialización como es el caso de grandes oclusiones de fondo o la aparición de objetos estáticos.
- *Clutter*. Categoría formada por secuencias que presentan grandes oclusiones del fondo mediante la aparición de grandes cantidades de objetos en movimiento.
- *Low frame rate*. Categoría sencilla similar a *Baseline*, con el inconveniente de estar formada por secuencias con poca cantidad de *frames*, provocando que la

[1] http://scenebackgroundmodeling.net/

estimación de movimiento se vea comprometida debido a los cambios bruscos que se producen entre *frames*.

- *Static Objects*. Categoría formada por secuencias que contienen objetos de frente que se encuentran más del 50 % de su duración sin movimiento, dificultando su diferenciación con el fondo de escena mediante un análisis temporal del movimiento de la misma.

SBMnet Dataset[2]. Dicho *dataset* contiene más categorías que el anterior. Está compuesto por más vídeos y estructurado en un mayor número de subsecciones (ver Figura 4.2), con las que se busca llevar a cabo más pruebas con el objetivo de encontrar los puntos fuertes y debilidades de los algoritmos propuestos. Estructuración de dicho *dataset*:

- *Background Motion*. Esta categoría está compuesta por vídeos que presentan un reto a la hora de realizar la inicialización del fondo, debido a la presencia de un fondo dinámico (p.ej. árboles, agua o escaleras mecánicas), complicando la separación entre frente y fondo mediante el movimiento.

- *Basic*. Categoría similar a *Baseline* del VPU *dataset*.

- *Clutter*. Categoría similar a *Clutter* del VPU *dataset*.

- *Ilumination Changes*. Esta categoría está compuesta por vídeos que presentan cambios de iluminación bruscos o graduales, provocando una estimación de fondo compleja en la cual el resultado puede estar compuesta por varios fondos.

- *Intermittent Motion*. Categoría similar a *Static Objects* del VPU *dataset*.

- *Jitter*. Categoría compuesta por vídeos que presentan un pequeño movimiento de la escena debido a la vibración de la cámara que graba la secuencia. Este movimiento de la escena, dificulta la diferenciación entre frente y fondo mediante el análisis temporal de la misma.

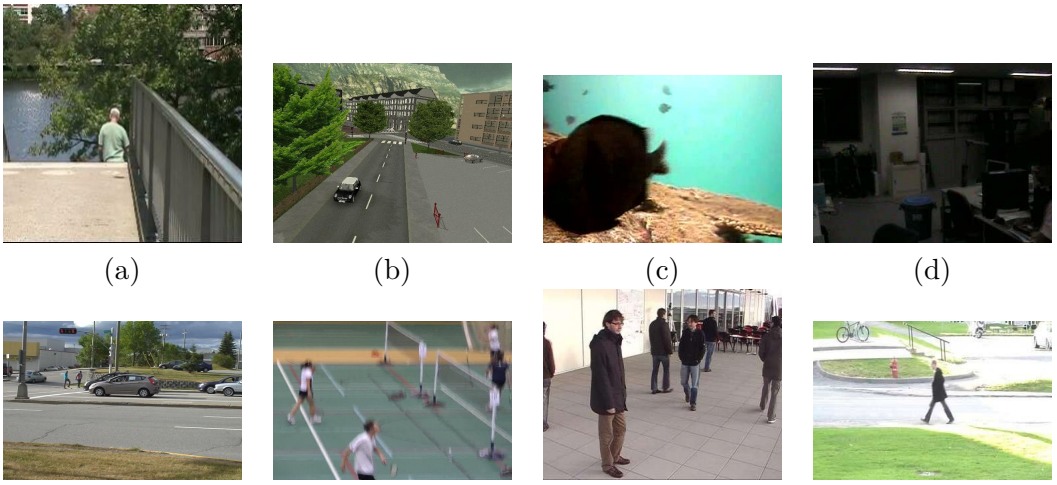
- *Very Long*. Categoría compuesta por secuencias de gran duración.

- *Very Short*. Categoría similar a *Low frame rate* del VPU *dataset*.

[2] http://scenebackgroundmodeling.net/

(a) (b) (c) (d)

Figura 4.1: Ejemplo de secuencias del VPU *dataset*. La imagen (a) hace referencia al apartado de *Basseline*, la imagen (b) a *Clutter*, la imagen (c) a *Low frame rate* y la imagen (d) a *Static Objects*.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figura 4.2: Ejemplo de secuencias del SBMnet *dataset*. La imagen (a) hace referencia al apartado de *Background Motion*, la imagen (b) a *Basic*, la imagen (c) a *Clutter*, la imagen (d) a *Ilumination Changes*, la imagen (e) a *Intermittent Motion*, la imagen (f) a *Jitter*, la imagen (g) a *Very Long* y la imagen (h) a *Very Short*.

4.1.2. Métricas

Para la evaluación de los resultados existen distintos tipos de métricas, algunas de ellas se encuentran explicadas en [29], las cuales se basan en la diferencia de luminancias, de color o el número de píxeles erróneos (con un margen de error) entre las imágenes de fondo generadas y los fondos óptimos de dichas secuencia aportados por los respectivos *datasets* denominados *ground-truth.*. Estas métricas son; *Average Gray-level Error* (AGE), *Percentage of Error Pixeles* (pEPs), *Percentage of Clusteres Error Pixeles* (pCEPs), *Peak-Signal-to-Noise-Ratio* (PSNR), *Color image Quality Measure* (CQM) y *Multi-Scale Structural Similaruty Index* (MS-SSIM). A pesar de que estos parámetros son los empleados por SBMnet para realizar la evaluación de algoritmos, se ha decido emplear únicamente el parámetro MS-SSIM para evaluar los

resultados del VPU *dataset*, debido a la similitud entre los resultados aportados con cada una de las métricas.

El *Multi-Scale Structural Similaruty Index* (MS-SSIM), es un parámetro que propone evaluar la calidad de la imagen de fondo generada en múltiples resoluciones o escalas (multi-scale) usando el método SSIM, el cual produce un mapa de puntuaciones comparando regiones y emplea la media para producir una medida de calidad final de la imagen. Su valor varía en el rango [0,1] y cuanto mayor sea este valor, mayor será la calidad del fondo generado.

4.2. Pruebas y resultados

En esta sección se realiza el análisis de los resultados obtenidos con la métrica de evaluación MS-SSIM tanto para el algoritmo base [3] y sus modificaciones. En primer lugar, se han estudiado los resultados obtenidos con cada uno de las modificaciones en el *dataset* del VPU. La selección de este *dataset* para realizar un análisis global se lleva a cabo debido a que dicho *dataset* aporta los *ground-truth*, necesarios para realizar la evaluación, mientras que SBMnet *dataset* no aporta sus respectivos *ground-truth* y requiere subir los datos a una plataforma de evaluación. Una vez obtenidos los resultados globales con todas las modificaciones, se realiza el análisis de las mejores configuraciones en el SBMnet *dataset*.

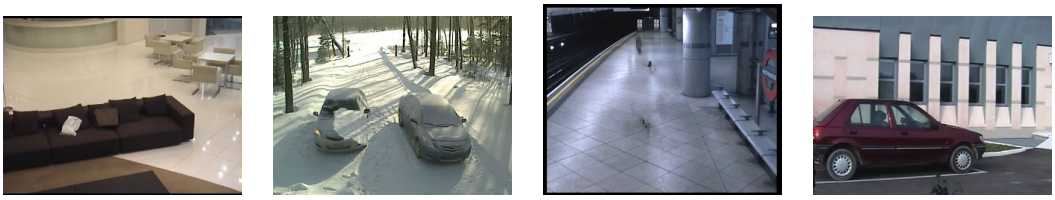
Con el objetivo de realizar una evaluación global de los algoritmos diseñados, se analizan los resultados del algoritmo base LabGen-P [3] (FDiff), explicado en la Sección 3.2, el cual se basa en el uso de *frame difference* para la estimación del movimiento. Las modificaciones implementadas sobre dicho algoritmo son las siguientes:

1. Modificación 1 (OF). Cambio de la estimación de movimiento empleada en el algoritmo original por el uso del flujo óptico definido en la Subsección 3.3.1.

2. Modificación 2 (Reg). Utilización de la información de tamaño de regiones como puntuación para diferenciar entre frente y fondo, desarrollado en la Subsección 3.3.2.

3. Modificación 3 (FDiff+Reg). Empleo de la información de movimiento mediante la estimación con *frame difference* y la información de tamaño de las regiones, como puntuación para diferenciar entre frente y fondo, desarrollado en la Subsección 3.3.3.

4. Modificación 4 (OF+Reg). Empleo de la información de movimiento mediante la estimación con el flujo óptico desarrollado en la Subsección 3.3.1 y la información de tamaño de las regiones, como puntuación para diferenciar entre frente y



(a) (b) (c) (d)

Figura 4.3: Ejemplos de fondos generados por el algoritmo base. Los fondos (a) y (b) se corresponden con secuencias con objetos estáticos y los fondos (c) y (d) con secuencias que presentan gran oclusión del fondo.

	FDiff	**OF**	**Reg**	**Reg+FDiff**	**Reg+OF**
Baseline	0.97445	0.97332	0.91351	0.95302	0.97503
Clutter	0.97096	0.97628	0.71628	0.83334	0.97284
LowFrame	0.97495	0.97844	0.97357	0.9756	0.97763
StaticObject	0.94991	0.92346	0.90283	0.95192	0.96364
MEAN	**0.9675675**	**0.962875**	**0.8765475**	**0.92847**	**0.972285**

Tabla 4.1: Tabla comparativa de los distintos algoritmos desarrollados con las medidas de MSSSIM para los distintos tipos de archivos del VPU *dataset*.

fondo, desarrollado en la Subsección 3.3.2, obteniendo el algoritmo desarrollado en la Subsección 3.3.3.

4.2.1. Resultados obtenidos con VPU *dataset*

El algoritmo desarrollado por LabGen-P, obtiene buenos resultados en muchos escenarios (ver Tabla 4.1), sin embargo está basado en una estimación de movimiento burda, como ya se ha explicado anteriormente (ver Subsección 3.3.1). Los resultados obtenidos en la mayoría de apartados es buena (ver Figura 4.3 (c) y (d)), pero cabe destacar la disminución del rendimiento en situaciones que presentan objetos estáticos (ver Figura 4.3 (a) y (b)). Por este motivo, el objetivo es mejorar el rendimiento en estas situaciones sin empeorar en el resto, para ello se usará la información de tamaños explicada en la Subsección 3.3.2. Una vez observada de manera global los resultados que se obtienen con cada algoritmo en la Tabla 4.1 donde se aprecia una ligera mejora en los resultados al emplear (OF+Reg), pasaré a analizar en detalle cada categoría.

4.2.1.1. Resultados de la categoría *Baseline*

Los resultados obtenidos en esta categoría se presentan en la Tabla 4.2 para cada uno de los vídeos que la componen. El mejor resultado se obtiene con (OF+Reg), pero al tratarse de secuencias que no proponen grandes retos a los algoritmos de

	FDiff	OF	Reg	FDiff+Reg	OF+Reg
CAVIAR1	0.9771	0.9451	0.9569	0.9737	0.9737
CAVIAR2	0.9993	0.9993	0.9991	0.9993	0.9993
CHUK	0.8477	0.8631	0.8585	0.8603	0.8616
HighwayI	0.9908	0.9906	0.9787	0.9806	0.9822
HighwayII	0.9948	0.9947	0.9931	0.9929	0.9938
HumanBody2	0.9910	0.9943	0.5135	0.7944	0.9920
Gate Traffic	0.9766	0.9746	0.8689	0.9604	0.9755
bankst	0.9858	0.9889	0.9873	0.9894	0.9900
Highway	0.9860	0.9873	0.9855	0.9843	0.9871
pedestrians	0.9954	0.9953	0.9936	0.9949	0.9951
MEAN	**0.97445**	**0.97332**	**0.91351**	**0.95302**	**0.97503**

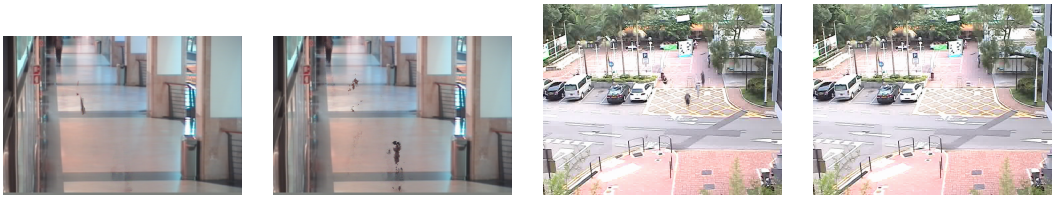
Tabla 4.2: Tabla con los valores MSSSIM para los archivos *Baseline* con todas las modificaciones implementadas.

inicialización de fondo, el resultado con la mayoría de los algoritmos es bueno. Cabe destacar, la disminución del rendimiento al emplear únicamente la información de tamaño de las regiones (Reg), la cual no es un gran método para inicializar el fondo por sí solo, debido a no emplear información de movimiento y basarse únicamente en la suposición de que los objetos más grandes serán los que formen el fondo.

Como ya se ha visto en la Subsección 3.3.1, la estimación de movimiento mediante flujo óptico es mucho más precisa, sin embargo esta diferencia no se aprecia notablemente en los resultados cuantitativos (ver Tabla 4.2). En las situaciones en las que se produzcan movimientos aislados los cuales no producen gran oclusión del fondo, la estimación mediante diferencia de *frames* se adapta bien a la escena, haciendo que las puntuaciones que son asignadas a zonas estáticas no influyan negativamente en el resultado, incluso obteniendo mejores resultados, como se puede apreciar en la Figura 4.4 (a) y (b), sin embargo en otras ocasiones en las que los movimientos son más difusos como en la Figura 4.4 (c) y (d), los resultados obtenidos son notablemente mejores tanto cualitativa como objetivamente.

4.2.1.2. Resultados de la categoría *Clutter*

Los resultados obtenidos en esta categoría en cada uno de sus vídeos se muestran en la Tabla 4.3. El algoritmo que obtiene mejores resultados es (OF) debido a la mejora en la calidad de la detección de movimiento en la secuencia. La diferencia con respecto a la otra estimación de movimiento es superior a la obtenida en la categoría anterior, debido a la aparición de objetos que producen gran oclusión de fondo, dificultando la diferenciación entre frente y fondo. El empleo de flujo óptico hace que



(a) (b) (c) (d)

Figura 4.4: Ejemplos de inicializaciones con (OF) y (FDiff) para la categoría *Baseline*. Las imágenes (a) y (c) se corresponden a fondos obtenidos con (FDiff) y las imágenes (b) y (d) a fondos obtenidos con (OF). Fondos de CAVIAR1, imágenes (a) y (b). Fondos de CHUK, imágenes (a) y (b).

(a) (b) (c)

Figura 4.5: Ejemplos de fondos para la secuencia Foliage con distintos algoritmos. La imagen (a) se corresponde con el fondo obtenido mediante (FDiff), la imagen (b) con (FDiff+Reg) y la imagen (c) con (OF+Reg).

la separación entre ambas informaciones sea más precisa, como se puede observar en la Tabla 4.3, este efecto proporciona unos resultados de mayor calidad. Igual que en el caso anterior, el empleo únicamente de información de tamaño de regiones para obtener el fondo de la secuencia, supone un gran deterioro de los resultados obtenidos. Este hecho se debe a que en esta categoría aparecen objetos de gran tamaño que ocluyen gran parte de fondo, por lo tanto, se seleccionan un gran número de regiones de frente como fondo.

Este efecto, hace que al introducir la información de tamaño junto a la de movimiento con *frame difference* el resultado empeore, ya que la puntuación de movimiento es alta en toda la imagen haciendo que la información de tamaño decida entre frente y fondo, y al ser esta mala por los motivos explicados anteriormente el resultado de (FDiff+Reg) empeora con respecto a (FDiff). Sin embargo con (OF) esto no ocurre debido a que al tener una estimación de movimiento precisa, se consigue que las zonas estáticas de fondo, tengan baja puntuación, entrando a formar parte del fondo de la escena. Este razonamiento se aprecia cualitativamente en la Figura 4.5. Por lo tanto, tiene más peso la mejora obtenida con el movimiento que el deterioro introducido por el tamaño de regiones.

	FDiff	OF	Reg	FDiff+Reg	OF+Reg
camera1-ps	0.9861	0.9832	0.9672	0.9843	0.9838
camera5-ps	0.9930	0.9926	0.9867	0.9923	0.9918
AVSS_AB_EVAL	0.9396	0.9581	0.8933	0.9448	0.9569
Board	0.9218	0.9489	0.7726	0.9058	0.9484
Crowd	0.9815	0.9835	0.9683	0.9700	0.9790
Foliage	0.9921	0.9922	0.0822	0.3976	0.9918
PeopleAndFoliage	0.9892	0.9909	0.1507	0.5087	0.9905
Campus	0.9715	0.9717	0.9546	0.9725	0.9747
Snellen	0.9631	0.9851	0.4920	0.6965	0.9824
bootstrap	0.9717	0.9566	0.8952	0.9609	0.9291
MEAN	**0.97096**	**0.97628**	**0.71628**	**0.83334**	**0.97284**

Tabla 4.3: Tabla con los valores MSSSIM para los archivos *Clutter* con todas las modificaciones implementadas.

	FDiff	OF	Reg	FDiff+Reg	OF+Reg
JapMall	0.9399	0.9449	0.9460	0.9460	0.9451
CreceNevado	0.9854	0.9855	0.9584	0.9781	0.9843
Payot	0.9765	0.9769	0.9729	0.9722	0.9744
Intersection	0.9894	0.9894	0.9894	0.9893	0.9894
Plaza Mayor	0.9638	0.9674	0.9679	0.9659	0.9665
ShoppingMall	0.9752	0.9786	0.9798	0.9669	0.9788
mall	0.9741	0.9789	0.9644	0.9751	0.9758
tramCrossroad	0.9678	0.9860	0.9834	0.9875	0.9864
tunnelExit	0.9859	0.9859	0.9841	0.9850	0.9851
turnpike	0.9915	0.9909	0.9894	0.9900	0.9905
MEAN	**0.97495**	**0.97844**	**0.97357**	**0.9756**	**0.97763**

Tabla 4.4: Tabla con los valores MSSSIM para los archivos *Low frame rate* con todas las modificaciones implementadas.

4.2.1.3. Resultados de la categoría *Low frame rate*

Del mismo modo que en el Subsubapartado 4.2.1.1, esta categoría esta formada por secuencias sencillas, en las cuales el reto reside en la gran magnitud de los movimientos que se producen entre los *frames* consecutivos. El algoritmo que mejor funciona es (OF) ligeramente superior al que emplea *frame difference* (FDiff) y con un rendimiento muy cercano al resto de algoritmos. Es importante destacar, la gran calidad de los fondos generados únicamente a partir del tamaño de sus regiones (Reg), debida a la naturaleza de los vídeos que forman esta categoría, por la cual las regiones de frente tienen un tamaño inferior a las de fondo.

	FDiff	OF	Reg	Reg+FDiff	Reg+OF
BusPolaco	0.9685	0.8244	0.9634	0.9805	0.9556
BusStopMorning	0.9656	0.9656	0.8961	0.9464	0.9420
CaVignal	0.9951	0.9675	0.7990	0.9978	0.9974
HallAndMonitor	0.9864	0.9875	0.8934	0.9869	0.9917
Gate Traffic	0.9695	0.9351	0.8305	0.8799	0.9731
abandonedBox	0.9830	0.9704	0.8944	0.9875	0.9840
granguardia	0.8368	0.8356	0.8748	0.8478	0.9039
office	0.9942	0.9700	0.9759	0.9913	0.9882
sofa	0.9846	0.9652	0.9593	0.9834	0.9793
winterDriveway	0.8154	0.8133	0.9415	0.9177	0.9212
MEAN	**0.94991**	**0.92346**	**0.90283**	**0.95192**	**0.96364**

Tabla 4.5: Tabla con los valores MSSSIM para los archivos *Static Objects* con todas las modificaciones implementadas.

4.2.1.4. Resultados de la categoría *Static Objects*

Los resultados de la categoría *Static Objects* se muestran en la Tabla 4.5. El objetivo de la introducción de la información de tamaño de regiones es mejorar en estas situaciones en las cuales la información de movimiento no se adapta a las circunstancias de los objetos de frente. En esta categoría el empleo de la estimación de movimiento mediante flujo óptico no resulta determinante en la calidad de los fondos obtenidos, es más, empeora levente respecto al otro modelo de estimación. Sin embargo con el empleo de ambas informaciones se obtiene una notable mejoría en los resultados, tanto cualitativa (ver Figura 4.6) como objetivamente (ver Tabla 4.5), gracias a la diferenciación de frente y fondo en situaciones dinámicas mediante la estimación de movimiento y la puntuación obtenida en situaciones sin movimiento con el tamaño de las regiones.

Por lo tanto, la información de tamaño de regiones nos aporta un punto de calidad en esta categoría (cuando los objetos de frente sean menores a los de fondo) con respecto al resto de algoritmos, siempre y cuando no perdamos la información referente al movimiento entre *frames*. Si los objetos de frente son mayores a los de fondo como en la Figura 4.6 (b), donde las piernas del hombre tienen un tamaño superior a la región del banco donde están situadas, el resultado será malo, pero no peor que el obtenido con la misma estimación y sin la información de tamaños.

(a) (b) (c) (d)

Figura 4.6: Ejemplos de inicializaciones con (OF) y (OF+Reg) para la categoría *Static Objects*. Las imágenes (a) y (c) se corresponden a fondos obtenidos con (OF) y las imágenes (b) y (d) a fondos obtenidos con (OF+Reg). Fondos de BusPolaco, imágenes (a) y (b). Fondos de Gate Traffic, imágenes (a) y (b).

4.2.2. Resultados obtenidos con SBMnet *dataset*

Una vez analizados los resultados obtenidos en el VPU *dataset*, he realizado la evaluación de los algoritmos (OF[4]) y (OF+Reg[5]) con el SBMnet *dataset*, debido a que son los que han obtenido mejores resultados, y con el objetivo de comparar nuevamente como influye la introducción de la información de tamaño de regiones en el algoritmo. Esta evaluación ha sido realizada en SBMnet[6].

Así pues, los resultados en este *dataset*, observables cuantitativamente en la Tabla 4.6 para (OF) y en la Tabla 4.7 para (OF+Reg), ponen de manifiesto que el algoritmo (OF) tiene un mayor rendimiento global. Cabe destacar que esta mejora en la inicialización de fondos cuantitativamente (ver Tablas 4.6 y 4.7) y cualitativamente (ver Figura 4.7 (c) y (d)) se produce mayormente en las categorías de *illumination changes, jitter* y *very long,* categorías para las cuales no se ha introducido ningún tratamiento especial. En el caso de *jitter,* la mejora se debe a que el empleo de regiones provoca un emborronado de la imagen final, debido a que selecciona como fondo aquellas imágenes que tengan regiones mayores, es decir las más borrosas (ver Figura 4.7 (d)). En el caso de los cambios de iluminación, el tamaño de regiones hace que en situaciones con igual movimiento, esta información regiones provoque más diferencias que espaciales bruscas, apreciables en la Figura 4.7 (c)).

Sin embargo en el caso del *clutter* y los objetos estáticos (ver Figura 4.7 (a) y (b))., se consiguen unos resultados de mayor calidad con (OF+Reg) como se puede observar en la Figura 4.7. Por lo tanto cabe esperar que si se implanta algún método para mejorar en situaciones de *jitter* y cambios de iluminación el resultado con (OF+Reg) sea mejor.

[4] http://pione.dinf.usherbrooke.ca/results/175/
[5] http://pione.dinf.usherbrooke.ca/results/174/
[6] http://scenebackgroundmodeling.net/

(a) static (b) static (c) (d)

Figura 4.7: Ejemplos de fondos generados en SBMnet *dataset* con (OF) y (OF+Reg). La fila superior se corresponde con los fondos generados por (OF) y la inferior con los generados por (OF+Reg). Las columnas (a) y (b) se corresponden con vídeos de la categoría *IntermittentMotion,* la columna (c) se con *Illumination Changes* y la columna (d) con *Jitter.*

	AGE	**pEPs**	**pCEPS**	**MSSSIM**	**PSNR**	**CQM**
Basic	4.0954	0.0170	0.0040	0.9739	32.9007	33.5827
IntermittentMotion	5.4719	0.0446	0.0259	0.9429	27.1628	28.1720
Clutter	6.8071	0.0652	0.0359	0.8921	29.7191	30.6948
Jitter	9.9310	0.1155	0.0420	0.8464	25.2217	26.3040
IlluminationChange	13.5119	0.1717	0.0955	0.8701	21.9988	23.3109
BackgroundMotion	9.4559	0.1218	0.0234	0.8658	25.9567	26.8510
Very Long	7.0840	0.0543	0.0119	0.9782	28.3124	29.2470
Very Short	5.5542	0.0393	0.0157	0.9476	30.0655	30.7724
MEAN	**7.7389**	**0.0787**	**0.0318**	**0.9146**	**27.6672**	**28.6169**

Tabla 4.6: Tabla con los resultados de (OF) en el SBMnet *dataset.*

	AGE	**pEPs**	**pCEPS**	**MSSSIM**	**PSNR**	**CQM**
Basic	4.0965	0.0172	0.0042	0.9746	32.6052	33.3074
IntermittentMotion	4.6078	0.0304	0.0156	0.9675	30.0102	30.8694
Clutter	5.1190	0.0334	0.0127	0.9345	29.6630	30.7031
Jitter	10.5245	0.1242	0.0486	0.8266	23.2106	24.3552
IlluminationChange	22.3650	0.2425	0.1772	0.8120	20.8067	21.9901
BackgroundMotion	9.7492	0.1253	0.0244	0.8607	25.1905	26.3785
Very Long	9.4518	0.1125	0.0441	0.9459	26.4389	27.4018
Very Short	5.6092	0.0397	0.0167	0.9466	30.0855	30.7938
MEAN	**8.9354**	**0.0906**	**0.0429**	**0.9086**	**27.2513**	**28.2249**

Tabla 4.7: Tabla con los resultados de (OF+Reg) en el SBMnet *dataset.*

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

En este TFG se ha implementado un algoritmo de inicialización de fondo basado en la mediana temporal desarrollado por LabGen-P, seleccionado por sus buenos resultados y sencillez. Con el objetivo de mejorar en algunos de los aspectos que este algoritmo no tenía en cuenta, se han introducido una serie de modificaciones sobre él.

En primer lugar se realizó un estudio sobre el estado del arte para conocer los algoritmos existentes en la literatura en este campo, así como el estudio de los retos que se presentan a la hora de realizar un algoritmo de este tipo. Después de realizar un análisis completo del algoritmo base, se introdujo un nuevo método para la estimación de movimiento y una nueva puntuación para diferenciar entre frente y fondo. Este nuevo *score,* está basado en el tamaño de las regiones que forman la secuencia con el objetivo de mejorar en la inicialización de fondo cuando aparezcan objetos estáticos en ella.

Tras la implementación de estas modificaciones, se realizó una evaluación de las mismas mediante el *dataset* proporcionado por el VPU con el objetivo de seleccionar el mejor algoritmo. Después de esta evaluación se seleccionaron los algoritmos basados en la estimación de movimiento mediante el flujo óptico (OF) y el que adicionalmente a esta estimaciión, introducía la información de tamaño de regiones (OF+Reg). A la vista de los resultados, se consiguió mejorar el algoritmo original con el empleo de estas dos informaciones, sin embargo, no resultó ser un algoritmo muy fiable, ya que los resultados dependían en gran medida del tipo de secuencia, en concreto de si los objetos estáticos presentes tenían un tamaño inferior a las regiones de fondo. Además, en un gran número de categorías se produce una bajada del rendimiento debida a la

introducción de esta información de tamaños. Comparando los resultados ubicados en la página de SBMnet, se observó que no existe ningún algoritmo que funcione correctamente en todas las categorías, ya que los que se centran en mejorar en alguna de ellas, empeoran en otras. Por último, tanto el algoritmo base desarrollado como cada una de sus modificaciones tienen limitaciones, ya que además de lo explicado anteriormente, son dependiente de si los objetos estáticos aparecen al principio o al final de la secuencia. Si aparecen al principio, no habrá problema siempre y cuando posteriormente al desplazamiento de estos objetos, se rellene mas de la mitad del *buffer* con fondo estático, ya que de esta forma con el uso de la mediana sobre dicho *buffer* se obtendría el fondo correcto, sin embargo, si estos objetos estáticos aparecen al final del vídeo, tendrán muchas probabilidades de pertenecer al fondo.

5.2. Trabajo futuro

Como se ha visto, la tarea de la inicialización de fondo todavía tiene un amplio margen de mejora, ya que ningún algoritmo funciona correctamente frente a todos los retos que existen en dicho campo.

La estimación de objetos estáticos con este algoritmo no es precisa, para mejorar en ella se podría realizar una mejor estimación de las regiones y sus bordes, ya que como ese puede apreciar en la Figura 4.6 (d), estos objetos no se eliminan totalmente debido a la imprecisa estimación de sus bordes. Para mejorar en este aspecto, también sería necesario e interesante introducir información espacial que aporte heterogeneidad al fondo de la imagen, es decir, que en ningún caso aparezcan partes del objeto de frente que resalten con el color del fondo.

Además de la mejora en este aspecto, también se podrían introducir mejoras en otras categorías como en los cambios de iluminación. Para solucionar este problema, se podría plantear un estudio de la luminancia media de un *frame* a otro, es decir, si la luminancia media cambia en gran medida de un *frame* a otro, se debería guardar uno y generan otro con los *frames* restantes.

Los escenarios en los que se emplea este tipo de algoritmos, sobre todo en el campo de la video-vigilancia, se encuentran en entornos con gran oclusión del fondo debido a que son empleadas en zonas públicas como centros comerciales, estaciones, etc. Por lo que resultaría práctico que todo este tipo de algoritmos sean más resistentes al *clutter*.

Bibliografía

[1] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer Science Review*, vol. 11â12, pp. 31 – 66, 2014. 1, 5, 6, 8

[2] L. Maddalena and A. Petrosino, "Background model initialization for static cameras," in *Background Modeling and Foreground Detection for Video Surveillance*, pp. 3–1, Chapman and Hall/CRC, 2014. 1, 2, 5, 6, 7, 8

[3] B. Laugraud, S. Pierard, and M. V. Droogenbroeck, "Labgen-p: A pixel-level stationary background generation method based on labgen," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 107–113, Dec 2016. 2, 7, 11, 14, 23, 26

[4] A. Colombari and A. Fusiello, "Patch-based background initialization in heavily cluttered video," *IEEE Transactions on Image Processing*, vol. 19, pp. 926–933, April 2010. 6, 7

[5] D. Park and H. Byun, "A unified approach to background adaptation and initialization in public scenes," *Pattern Recogn.*, vol. 46, pp. 1985–1997, July 2013. 6, 8

[6] H.-H. Hsiao and J.-J. Leou, "Background initialization and foreground segmentation for bootstrapping video sequences," *EURASIP Journal on Image and Video Processing*, vol. 2013, no. 1, pp. 1–19, 2013. 6, 7

[7] L. Maddalena and A. Petrosino, "The sobs algorithm: What are the limits?," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 21–26, June 2012. 6, 7

[8] V. Reddy, C. Sanderson, and B. C. Lovell, "A low-complexity algorithm for static background estimation from cluttered image sequences in surveillance contexts," *CoRR*, vol. abs/1303.2465, 2013. 6, 7

[9] R. Zhang, W. Gong, A. Yaworski, and M. Greenspan, "Nonparametric on-line background generation for surveillance video," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 1177–1180, Nov 2012. 6, 7

[10] R. M. Colque and G. CÂŽmara-ChÂŽvez, "Progressive background image generation of surveillance traffic videos based on a temporal histogram ruled by a reward/penalty function," in *2011 24th SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 297–304, Aug 2011. 6

[11] T. Crivelli, P. Bouthemy, B. Cernuschi-Frías, and J.-f. Yao, "Simultaneous motion detection and background reconstruction with a conditional mixed-state markov random field," *International Journal of Computer Vision*, vol. 94, no. 3, pp. 295–316, 2011. 6, 8

[12] D. Ortego, J. C. SanMiguel, and J. M. Martínez, "Rejection based multipath reconstruction for background estimation in video sequences with stationary objects," *Computer Vision and Image Understanding*, vol. 147, pp. 23–37, 2016. 6, 7, 8, 16

[13] H. L. Eng, K. A. Toh, A. H. Kam, J. Wang, and W. Y. Yau, "An automatic drowning detection surveillance system for challenging outdoor pool environments," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 532–539 vol.1, Oct 2003. 7

[14] L. Maddalena and A. Petrosino, "The 3dsobs+ algorithm for moving object detection," *Computer Vision and Image Understanding*, vol. 122, pp. 65 – 73, 2014. 7

[15] C. C. Chen and J. K. Aggarwal, "An adaptive background model initialization algorithm with objects moving at different depths," in *2008 15th IEEE International Conference on Image Processing*, pp. 2664–2667, Oct 2008. 7, 8

[16] V. Reddy, C. Sanderson, and B. C. Lovell, "An efficient and robust sequential algorithm for background estimation in video surveillance," in *2009 16th IEEE International Conference on Image Processing (ICIP)*, pp. 1109–1112, Nov 2009. 7, 16

[17] H. Wang and D. Suter, "A novel robust statistical method for background initialization and visual surveillance," in *Proceedings of the 7th Asian Conference on Computer Vision - Volume Part I*, ACCV'06, (Berlin, Heidelberg), pp. 328–337, Springer-Verlag, 2006. 7

[18] D. Baltieri, R. Vezzani, and R. Cucchiara, "Fast background initialization with recursive hadamard transform," in *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, pp. 165–171, Aug 2010. 7

[19] A. Shrotre and L. J. Karam, "Background recovery from multiple images," in *Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE), 2013 IEEE*, pp. 135–140, Aug 2013. 7

[20] X. Xu and T. S. Huang, "A loopy belief propagation approach for robust background estimation," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–7, June 2008. 8

[21] C. Guo, S. Gao, and D. Zhang, "Belief propagation algorithm for background estimation based on local maximum weight matching," in *Image and Signal Processing (CISP), 2012 5th International Congress on*, pp. 82–85, Oct 2012. 8

[22] B. Laugraud, S. Pierard, and M. V. Droogenbroeck, "Labgen: A method based on motion detection for generating the background of a scene," *Pattern Recognition Letters*, 2016. 11

[23] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, *High Accuracy Optical Flow Estimation Based on a Theory for Warping*, pp. 25–36. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. 14, 15

[24] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 568–576, Curran Associates, Inc., 2014. 15

[25] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 15

[26] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. SÃŒsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 2274–2282, Nov 2012. 16

[27] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *ICCV*, 2013. 16

[28] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 2294–2301, IEEE, 2009. 16

[29] L. Maddalena and A. Petrosino, *Towards Benchmarking Scene Background Initialization*, pp. 469–476. Cham: Springer International Publishing, 2015. 25